

Issues and strategies for the physical design of system-on-a-chip ASICs

by T. R. Bednar
P. H. Buffet
R. J. Darden
S. W. Gould
P. S. Zuchowski

The density and performance of advanced silicon technologies have made system-on-a-chip ASICs possible. SoCs bring together a diverse set of functions and technology features on a single die of enormous complexity. The physical design of these complex ASICs requires a rich set of functional elements that integrate efficiently with a set of design flows and tools productive enough to meet product requirements successfully, without consuming more time or design resources than a simpler design. The architecture described, including functional libraries and physical design conventions, enables the creation of multiple SoC ASIC designs from a common infrastructure that addresses silicon integration, electrical robustness, and packaging challenges. An implementation strategy follows from this design infrastructure that includes hierarchical design concepts, placement, routing, and verification processes.

Introduction

An ASIC is a chip with application-specific function, designed using predefined elements from a circuit library,

assembled, interconnected, and timed by a design system. Companies which provide ASIC design systems and services construct ASIC products targeted to a specific silicon technology, and provide the product to their customers in the form of a design kit.

The ASIC design style provides the design customer a turnaround time (TAT) and design resource advantage over full-custom transistor-level design. The challenge for the ASIC provider is to limit the impact to cost and performance by the use of standard, reusable ASIC elements in contrast to the flexibility of full-custom design.

The combination of high-density silicon processes and advances in ASIC design system software capabilities have made custom system-on-a-chip (SoC) design possible. An SoC is an ASIC that integrates, on a single silicon die, processors, memories, logic, analog, and I/O functions previously implemented as multiple discrete chips. Some of these predefined functions are offered as “hard cores,” which are complete physical implementations of circuits and interconnects in a fixed form factor. A function offered as a fully defined logical netlist of a standard function that can be uniquely physically configured and wired in a specific implementation is called a soft core.

The evolution of an ASIC from custom logic to complete SoC presents the ASIC design system with new issues and challenges. The underlying silicon technology must have a more complex, diverse feature set to support

©Copyright 2002 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/02/\$5.00 © 2002 IBM

Table 1 IBM CMOS8sf transistor offerings.

<i>Transistor</i>	<i>Delay (ps)</i>	<i>Oxide thickness (nm)</i>	<i>Off-current (nA/μm)</i>
Standard n-MOS	18.5	2.2	0.16
Low- V_t n-MOS	14	2.2	4
High- V_t n-MOS	30.5	2.2	<0.01
MPU n-MOS	12	1.7	10

the varied functions characteristic of an SoC. Given the enormous range of designs possible, customization of each component element is not practical. Very complex hard and soft cores can be defined, verified, and reused on multiple SoC ASICs without customization. The issues related to integrating these large fixed blocks are much different from those involving the integration of primitive logic elements in traditional standard cell design.

This paper describes strategies for addressing the challenges of SoC design from two perspectives. First, the architecture of the SoC ASIC design system is considered, including how to integrate and optimize technology features for a large number of specific designs with different requirements and content. A menu of die size options and package offerings that satisfies a wide variety of SoC applications is also required. The physical, thermal, and electrical attributes of the die and package combination are critical considerations for complex applications. In many ways, designing an SoC package and die image is as complex as the card design of the previous generation.

Second, this paper discusses the physical design requirements of a specific SoC implementation. Floorplanning is a key initial step in the physical design of any SoC ASIC. The choice to use a hierarchical design methodology instead of a single-level, flat style is a critical decision when designing an SoC. I/O planning and placement can be problematic in SoC design, given potential interactions with other design blocks, and electrical interactions with on-chip power distributions and package characteristics. In an SoC environment, the I/O and package interactions are made more complicated by the existence of multiple voltages. On applications of this complexity, a comprehensive electrical analysis is required. When all of the blocks are planned and placed, detailed power, signal, and clock routing can take place. Finally, logic, physical, and electrical checking can be completed on the resulting design.

Throughout the process of both architecting the ASIC design system and designing the specific SoCs themselves, cost, performance, and TAT are the key attributes that are targeted for optimization. The primary goal of the design

system is productivity—the ability to optimize a specific application from a reusable set of functional circuit elements, silicon, and package technology, and to perform that optimization with a minimum of design resources and design time. The single most effective way to realize this productivity gain is to maximize the probability of first-time success, thus eliminating the need for costly and time-consuming multi-pass designs.

SoC design system architecture

The ASIC design kit contains the base circuit elements (library), software, and packaging options that are required for the design and construction of an SoC design. Because the ASIC design kit may be used as the basis for hundreds of unique designs, it must not only be optimized for performance and end-product cost, but must also be architected to minimize design TAT. To this end, many architectural decisions are made during the design of the ASIC design kit. These decisions will ultimately affect what is possible in an SoC design. In this section, we examine three of these choices: the technology design point, in terms of transistor choice and metal-level stack; the circuit library content required by SoC design; and the image/package combinations supported. We consider the Cu-11 ASIC design system, which uses CMOS8sf, the IBM 0.13- μ m semiconductor process, as an example of the decisions that are made, how they are made, and their impact on SoC designs.

Technology features

The high level of functional integration demanded by an SoC design involves the integration of silicon features that were previously optimized in separate technology offerings. The distinct functional blocks were individually optimized for their own relatively narrow, functional requirements. In the SoC world, these features must be combined in an economically viable way, and this may generate technical tradeoffs and challenges.

Today's leading semiconductor processes offer a number of transistor options. The choice of transistor(s) for use in the ASIC library is one of the most important architectural decisions made during the creation of the library, because this choice will affect the performance, cost, and power attainable on chip designs which utilize the library. **Table 1** presents the CMOS8sf logic n-MOS choices. The table lists the various transistor options; the delay obtained with a sample circuit utilizing each transistor option; gate-oxide thickness; and off-current (leakage current) of each of the transistor options.

As the table shows, the CMOS8sf process offers a wide range of transistors with various delay/off-current ratios. The choice of the “right” transistor depends on the end use of the SoC. For example, low-leakage-battery-operated parts must use the high- V_t n-MOS with its low I_{off}

Table 2 IBM Cu-11 metal-level options.

<i>Metal layers</i>	<i>0 thick levels</i>	<i>1 thick level</i>	<i>2 thick levels</i>	<i>3 thick levels</i>
3	3 thin + 0 thick			
4	4 thin + 0 thick	3 thin + 1 thick		
5	5 thin + 0 thick	4 thin + 1 thick	3 thin + 2 thick	
6	6 thin + 0 thick	5 thin + 1 thick	4 thin + 2 thick	
7			5 thin + 2 thick	4 thin + 3 thick
8			6 thin + 2 thick	5 thin + 3 thick

characteristics, while performance-critical designs would opt for the MPU device for its improved delay. Unfortunately, the intellectual property (IP) components are often designed and qualified months in advance of the SoC. Development budgets prevent the creation of IP optimized for each design point, so the ASIC architect is forced to choose which transistor option to support before knowing the end application.

When selecting a transistor option for the Cu-11 ASIC library, the architects consider performance gains from the previous product, aiming for a 20% improvement from technology to technology. The team also considers cost implications (wafer processing, mask, and test) as well as the power requirements of known or anticipated customers. In this example, a decision is made to build a product around the standard n-FET shown in Table 1 because the high- V_t device offers outstanding leakage attributes but does not meet the 20% performance improvement objective, while the low- V_t device offers excellent performance but the leakage current is excessive for low-leakage applications. The use of this transistor option requires the use of an intermix methodology in which two different transistor options are used on the same design in an effort to balance performance and power objectives. While the MPU n-FET is 35% faster than the standard n-FET and would offer outstanding performance, the performance comes at the price of 62 times the leakage current and much more process complexity due to the different gate-oxide thickness. The use of the MPU transistor by itself is not possible for most SoC customers because of power considerations. Another important design criterion for an SoC is the number of metal layers (also called the metal stack) used in the SoC chip design. It is desirable to use the fewest possible levels of metal to complete the SoC design in order to minimize cost. Like the transistor choice, the metal-level options available to the SoC designer are set by the ASIC design kit. Cu-11 offers the metal-level options listed in **Table 2**.

Thin levels of metal are defined as those levels of metal that offer the tightest wiring pitch (width plus space) available in the technology. Thick levels of metal are defined as metal levels that have a larger cross-sectional area (both in thickness and width) and therefore offer

superior performance and current-carrying capability at the expense of a looser wiring pitch. Thin levels offer more wiring channels per unit area and help to minimize die size and cost. Thick levels of metal are better for power distribution, high-speed clocking, and routing of signals that must travel long distances. An additional constraint on the metal stack is that once a hard core is routed, the metal levels used are fixed and cannot change without significant redesign. It is important that the ASIC design kit specify the metal stack options that can be used for routing hard cores so that different hard cores do not employ incompatible metal stacks. This metal stack compromise can sometimes result in a hard core that is larger than absolutely necessary. This tradeoff is generally accepted as the price required to allow SoC designs to be designed quickly using existing IP and the ASIC design system.

The Cu-11 ASIC product chose to support the metal-level options shown in boldface in Table 2. The architects felt that two thick levels of metal were always required for power distribution and high-speed clock routing. With four levels of metal reserved for hard cores, the minimum metal stack configuration in Cu-11 is six levels: four thin, two thick.

SoC circuit library considerations

Logic libraries

The ASIC library is used to implement both the custom logic in an SoC design and the logic in the hard and soft cores included in the ASIC design kit (for example, the embedded processors). To provide a time-to-market advantage to the SoC designer, the ASIC library must be robust, pre-characterized, hardware-verified, and designed for ease of use.

Extensive simulation is performed on each library element at process and environmental extremes. The library is typically characterized at the 3σ process limits to allow for the largest manufacturing process window. Hardware-to-model correlation is performed on the library to ensure that the timing modeled in the ASIC design kit is representative of the actual timings seen in hardware. Hardware reliability stresses (exposure of the die to high

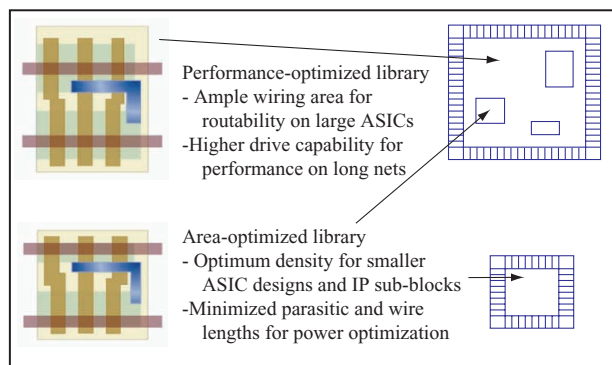


Figure 1

SoC logic library applications.

voltage and temperature) are performed on the library to minimize the risk of reliability problems on later designs. Although these techniques result in conservative designs, this is considered a reasonable tradeoff for avoiding hardware functionality issues in SoC designs which must have a short time-to-market.

The IBM Blue Logic* Cu-11 ASIC product offers SoC designers two distinct logic library options: a high-density 9-track logic library, and a high-performance 12-track logic library. The logic libraries differ only in physical size and maximum drive capability. The libraries are equivalent in functional content and have similar layout topologies. The difference between the two libraries lies in the size of the unit cell. The 12-track library is built in units of 12 global wiring tracks high by one track wide, while the 9-track library unit cell is 9 tracks high by one track wide.

The base 12-track Cu-11 library, like libraries in previous IBM technologies [1], offers high drive capability and supports ample wiring resources over the cell to maintain high levels of routability for large and high-performance designs. This library design point has proven to be robust over a large range of die sizes. The drive capability of the larger cell also allows efficient drive for the long nets in large, high-performance SoC designs.

The smaller physical size of the 9-track library can better optimize density for smaller ASIC designs, IP sub-blocks, and cores. In smaller designs (Figure 1) [2], where placement density is not driven by wiring congestion, the finer granularity of the 9-track library can provide a density advantage and can minimize wire lengths and parasitics for power optimization. Designs have shown up to 15% area reduction depending on logic content when the 9-track library is used at both the chip level and sub-block/core level.

Hierarchical design is used when mixing the 9-track and 12-track libraries on the same chip. Random logic macros

(RLMs) are defined to separate 9-track and 12-track hierarchical entities. SRAMs and other hard blocks are not affected by this logic library option, and can be placed in either 9-track or 12-track terrains. Gate array libraries are available in both 9-track and 12-track versions for backfill and metal-level personalization changes.

Embedded DRAM

Another major class of building blocks that must be integrated into an SoC are high-density memory macros. In the Cu-11 product, high-density trench DRAM macros can be integrated on SoC ASICs in various quantities. Competitive DRAM designs require some specific technology features that are historically distinct from pure digital logic processes. In addition to technology, DRAM designs have evolved specific design and test implementations which complicate the delivery of high-density embedded DRAM memories [2, 3].

From a technology standpoint, the integration of a trench capacitor is the most difficult integration issue for successfully embedding DRAM without affecting the integrity of the high-performance logic technology. The trench processing is lengthy, and it must be done before fabricating transistors. Lengthy wafer-processing TATs might be acceptable in standard products, but they are a key market detractor for a custom SoC design. To mitigate this issue, the placement of the DRAM macros can be done early in the design process, and frozen. The mask design levels specific to the trench can then be captured, masks made, and wafer processing started through the trench sectors in parallel with the remainder of the SoC physical design process. This process can effectively remove the TAT impact associated with the embedded trench DRAM.

Embedded DRAM designs also typically require design and technology features that are not common to pure logic technology. In an effort to maintain acceptable performance, reliability, availability, and retention times, techniques such as local voltage boosting are often used in DRAM macros. Maintaining low leakage currents is key to meeting these important functional specifications, and it requires techniques such as well biasing and the use of thick oxide transistors to limit tunneling currents. These requirements and techniques use some of the same well-isolation and multiple-voltage approaches employed to support analog design, but are generally contained within the confines of the fixed memory macro. The technology implications of embedded DRAM designs are further detailed in a companion paper in this issue [4].

The inclusion of 10–100 Mb of embedded DRAM in an SoC ASIC design can add a tremendous amount of densely packed, defect-sensitive area to the die, which could seriously limit the manufacturing yield. For a long time, standalone DRAMs have had designed-in,

sophisticated redundancy schemes to allow bits with manufacturing defects to be swapped out of the address space of the array. Multiple banks of redundant word and bit lines are utilized to provide a wide range of fixable memories. Fuse options are required as a technology feature to provide the nonvolatile programmability required to affect this redundancy. The traditional technology for redundancy programming has been laser-blown fuses, which present several issues to SoC physical design. Laser fuses must reside at the top level of metal in order to be accessed by the laser energy, and must have ample space around and beneath them to prevent damage to surrounding circuitry during the blow process. This presents significant placement challenges for the fuse block. While the memory may be implemented with only several levels of metal, allowing the additional metal levels in the stack for wiring the rest of the ASIC, the fuse block must consume all of the metal levels for the fuse to be implemented safely. This not only represents wiring blockage to the rest of the die, but also may conflict with the placement of signal and power pads on the die. This has caused the fuse-block portion of the redundancy circuitry to be moved away from the memory macro itself, so as to free the macro from the placement difficulties of the fuses. The fuse blocks are now independently placed, smaller objects to ease these floorplanning issues with dense memories [5].

As technologies scale to greater densities, metal levels increase, and low- k dielectrics are introduced, the relatively constant size and energy required to blow laser fuses will make them increasingly unattractive. Electronically blown fuses (anti-fuses) will replace laser fuses and remove many of the metal-level and placement conflicts described above. However, these electronic fuses present different integration issues. They require elevated voltage and current levels to be blown, and this drives the architecture of a mechanism to steer these fuse currents to the proper links from an external source.

The fuse methodology developed for embedded DRAM can also be used for SRAMs or other functional structures, to implement an electronic chip ID that gives each manufactured die a unique signature for traceability, and to store manufacturing or test data on the die itself.

Analog integration

Analog functions integrated on an SoC are one example of a set of functions previously available at the system level in function-specific technology which must now be integrated in silicon with digital components. Analog design often requires a very different variety of technology features, model accuracy, and integration sensitivities than digital design. The device set common to digital and analog design (primarily transistors, resistors, and diodes) can readily be used by both styles of design, but the

sensitivity to parasitics and overall modeling accuracy is much higher in analog design. In addition to the set of elements held in common with digital technology, another set of design elements is required for extensive analog design. These elements are made to be modular extensions to the base digital technology; i.e., they can be added to the fabrication process without affecting the characteristics of the other elements in the technology. Elements in this category, in which processing steps are added specifically to build the elements, include metal capacitors, inductors, and precision resistors. The modularity of these elements allows them to be employed only when they are valuable to the end application; the base elements that do not require these features can be used with or without these additional process steps. In other cases, analog design elements are structures that can be created from fundamental digital processing steps, but require more detailed characterization and modeling to be useful in analog design. These elements include diodes, thin oxide capacitors, and various resistive elements.

Analog integration on an SoC introduces a new set of electrical issues. Modeling of noise coupling is a critical need in analog design, and requires the extraction of substrate and well characteristics not typically required in most digital designs. A resistive substrate connection is essential for good noise isolation for analog designs, but requires more stringent design rules to prevent latch-up in digital designs. Noise sensitivity and isolation require considerable special handling of analog signals as they traverse the SoC, outside of the analog function itself. These signals cannot tolerate much series resistance, so they are wired using thick metal to reduce inductance and IR drops, and maintain acceptable electromigration lifetimes for high-current nets.

The most problematical requirement of analog integration is the need for power-supply voltages which are at different potentials and/or electrically isolated from digital power supplies. Many analog functions require voltages higher than the ever-shrinking digital logic voltages, forced by either long-existing standards or headroom to distinguish voltage steps. As a result, many analog functions require a unique off-die power-supply connection. These unique power connections are termed " AV_{dd} " connections to denote their treatment as analog power supplies. The placement of these analog blocks and the routing of the analog power-supply connection are critical components of the die-planning process.

There are design-tool and customer issues related to analog integration as well. Many fundamental design and verification tools used in ASIC design have been optimized over the years for efficient digital design. Many design tools commonly used for analog design are optimized for transistor-level design and are not efficient for multi-million-gate designs. Some of these tool conflicts

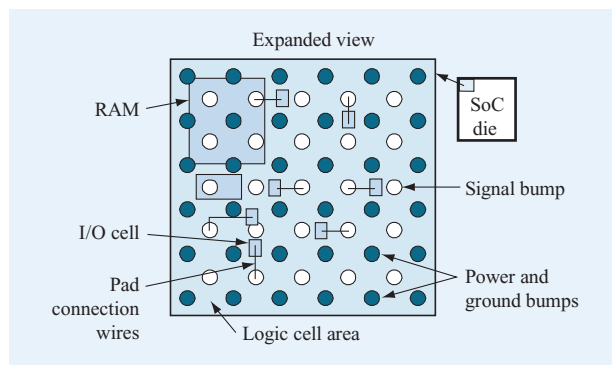


Figure 2

SoC placement flexibility of I/Os and logic, with area array image. Reprinted with permission from [6].

are subtle and fundamental (for example, the number of terminals on a transistor, and whether the well connections of a transistor are explicit or implied globally). Many of these differences can be abstracted at the block level to allow die-level integration, but they complicate device-level verification methodology and limit the exchange of transistor-level design IP between SoC designs.

Many of the design-tool issues can be insulated from the SoC designer if the analog function is contained within a fixed “black box.” Standard analog functions can be made available as fixed, off-the-shelf core macros. However, many analog functions, particularly higher-frequency designs, must be tuned to the specific application and the surrounding physical and electrical environment. In these cases, the end customer must have access to transistor-level design of the embedded analog function. For this reason, a common design environment and tool set has been developed to enable end-customer design and tuning of these sensitive circuits.

High-speed links

SoCs that require high serial data rates between SoCs call for advanced package and die designs. For easier signal escape and wiring through the package, high-speed links (HSLs) are placed near the die periphery. The solder interconnects (bumps) needed to service an HSL are in a fixed position with respect to its footprint. An HSL, or a group of HSLs designed as a single block, requires its own power, voltage reference, and signal interconnects, with minimum on-die routing. Time references such as PLLs are often dedicated to the HSLs, and also have pre-assigned signal interconnects.

As SoCs with several hundred HSLs emerge, they require placement flexibility for efficient use of the die

area. Groups of four, eight, or sixteen HSLs are designed without gaps between the HSLs they include. These HSL groups can be stacked laterally or vertically. The HSL groups do not interrupt the power busing because “power rings” are placed around the HSL group, to offer a similar power-grid resistance with and without the HSL group. To reduce the signal wiring blockage formed by the HSL groups, the porosity is enhanced with tracks through the groups, allowing non-HSL signals to pass through the macro.

The symmetry of the die footprint dictates how the HSL groups can be placed on the die, with an orientation suitable for signal fan-out through the package. For unrestrained HSL group placement, it may be necessary to design one to eight different HSL group bump patterns. Accordingly, several unique HSL group designs may be required.

Package and electrical considerations

The die “image” consists of metal buses and terminals that connect the die to the package: These are either wire-bond pads or solder-bump interconnects. With an SoC design, power must be distributed on-chip to each of the hard cores and to the standard cell logic on the die. The package design process can be time-consuming, and the test infrastructure is expensive and requires a long lead time. The ASIC design kit solves these problems by relying on fixed die sizes and predefined packaging solutions. These predefined solutions can help to minimize the application-specific electrical analysis required and the expense and time required to generate a packaged solution.

The ASIC flip-chip offerings are designed around a common solder-bump footprint supporting a number of different packaging options, allowing the wafer test infrastructure to be shared. In addition, IP which is required to interact with the solder bumps can be designed once to support a wide range of applications.

Image styles

In “peripheral” images, the wire-bond pads or the solder bumps are placed at the die periphery in order to facilitate the package signal wiring and to have fewer package wiring layers. The I/Os are located near the pads, or bumps, at the periphery of the die, in order to minimize the on-die connection distance [1].

The “area array” images offer complete placement flexibility for the I/Os and for the other logic elements (Figure 2). Signal bumps may be placed over the complete die area, and their density can be selected to fit a specific application (Figure 3). Specialized I/Os are accommodated by using a variety of power supplies that are “ V_{ddx} ”-compatible with the various industry standards for signaling. Up to four different V_{ddx} voltages are offered, each V_{ddx} available on a different quadrant of the die.

The placement, routing, and verification tools used in conjunction with area array images are more complex because of the higher degrees of freedom and the multitude of possible placements.

Package and image electrical attributes

Together, the package and the die image incorporate the power supply of the die. The dc currents follow paths from the board through the package and the image to the transistors on the die. The dynamic current paths are different. Some charge is stored by on-package decoupling capacitors, some charge is stored by on-die decoupling capacitors, and the rest of the current propagates from the board. The power-supply behavior is quantified with the help of detailed models and simulation at the frequencies of interest.

In addition to supplying the die with dc currents, the package and the image power-supply structures are the references for the signals traveling through the package and on the die. The typical cross section of a package is a triplate, with the signal wires sandwiched between reference planes. The coupling of the signal wires to their reference planes forms a return path for the signal dynamic current. A closed loop is needed for a complete current return path.

To quantitatively evaluate and compare the packages considered for an SoC, modeling and simulation are often necessary. The model extraction techniques are reviewed in [7]. Modeling is a complex task; the model matrix for a contemporary package is large, and the simulations for large-configuration computers require days. Also, the simulation results can be uncertain because of uncertainties in modeling techniques.

Rather than using large extracted models, smaller special-purpose models are often derived [8–10]. Frequency-domain analysis of the power supply separates the contributions of the die, the package, and the board resistance, inductance, capacitance (*RLC*) components to the power-supply response to switching activity. In a simulation of the SoC, such models help answer the SoC designer's power-supply integrity questions relative to the die power grid and its decoupling capacitors, the package inductances and decoupling capacitors, and the board regulator and its decoupling capacitors.

Specific SoC implementation

Physical design hierarchy

Two fundamental approaches are employed for the physical design of an ASIC: “flat” or hierarchical design. In a flat design methodology, all of the primitive and macro-level elements are placed, wired, and timed together at the die level. In a hierarchical approach, the design is partitioned; portions are designed independently

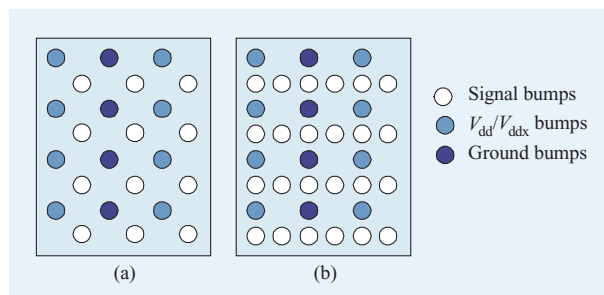


Figure 3

Area array bump patterns: (a) single- and (b) double-density signals. Reprinted with permission from [6].

and then integrated at the top level for the complete die. SoCs most often require a hierarchical approach in order to efficiently integrate multiple fixed functional elements of various types. Several types of hierarchy are supported by the IBM Blue Logic ASIC physical design methodology.

Move bounds are areas defined on the chip to which the placement of certain logic has been restricted. These are nonexclusive areas on the chip: The restricted logic must be placed in the area, but other logic is not excluded from it. If the placement software indicates that an unrestricted gate should be placed within a move bound, it is free to place it there. Since move bounds are area restrictions, the logic within all move bounds is placed and routed at the same time along with any remaining unrestricted logic.

In the IBM design system, random logic macros (RLMs) represent levels of hierarchy in the physical design domain. Each RLM has a size and shape, and all RLM logic must be placed within the boundary of the RLM. Logic from other portions of the design cannot be placed within that boundary. RLMs are placed and routed separately from one another, and from the top level. Possible interactions between wires at the RLM level and the top level may introduce dependencies (hence introducing an ordering to the placement and routing processes).

When to leverage hierarchy

The most dominant metric for design size is the number of placeable objects. The number of nets in a design is approximately equal to the number of placeable objects. Most physical design programs have CPU and memory requirements proportional to the number of placeable objects. Computing requirements of wiring programs are also proportional to the die size (in square mm).

IBM experience with the current generation of physical design tools shows that physical design TAT can be

improved by using hierarchical design techniques when the number of placeable objects reaches about 1.1 million. This is not a hard limit of the physical design programs (data volume test cases have been completed on designs in excess of 6.6 million objects), but rather a point at which hierarchical design should begin to be considered.

Today's SoC often consolidates the function of multiple chips in earlier technologies, optionally adding new function. The use of hierarchical physical design allows the mature portion of the design to be isolated from the new logic. The physical design on the mature portion can be completed while the new logic is designed. Once the new logic is complete, the physical design can be performed on it, and it can be merged with the existing design and quickly released to manufacturing. The size of the physical design team is also an important consideration. A hierarchical design methodology makes it much easier to use more than one physical designer on a design.

Another method used to fill large-capacity chip images is to place multiple copies of a design (or portion of a design) on a single chip. This is often seen in communication designs, where many copies of HSLs are placed on a single chip. The physical design on this macro can be performed just once and reused for each of the other instances, significantly reducing the total amount of physical design required for the chip.

The costs of hierarchical physical design

Hierarchical physical design has associated with it several costs that must be weighed against the potential benefits just described when making the flat-versus-hierarchical decision.

Hierarchical designs consume more resources than flat designs of the same size because of unique steps in a hierarchical methodology. Typically, one full-time physical designer is assigned to chip-level design, integration, and data management, and one or more part-time physical designers are assigned to focus on RLMs.

In hierarchical designs, a flat design is first executed on each of several RLMs, which are then integrated at the top level. This requires data management, which consumes resources that would not be required with the use of only a flat chip methodology.

Additional complexity is introduced because RLMs and the chip level share chip resources, such as wiring planes. When one entity (such as an RLM) uses a portion of a wiring plane, the used portion must be marked as blocked to all other entities (such as the chip level). Organizing and keeping track of all these blockages results in a process-management role that must be performed.

Placement, optimization, and wiring programs are hindered by hierarchical boundaries, resulting in designs

that are often less optimal than equivalent flat designs. Hierarchical boundaries represent walls to optimization programs that cannot "see" into or out of the RLM. This leads to decisions that are often locally optimal but globally suboptimal. A good example of this is clock optimization: Clock skew within an RLM is often very good; however, it is more difficult to achieve good clock skew between RLMs. This leads to an overall larger chip-level clock skew, but within the RLM, where the large majority of flip-flop connections exist, the clock skew is lower than that which could be achieved using a flat-chip approach.

Chip planning

SoC designs require extensive planning when architecting the chip physical layout. The large number of hard cores on many SoC designs, in addition to the numerous voltages required for interfaces, present the physical design team with many partitioning challenges. An important consideration for the design team is the partitioning of voltage on the design. Each supply that is required on the design creates extensive overhead in card wiring, packaging, on-chip electrostatic discharge (ESD) protection, on-chip level shifting, and on-chip power distribution; therefore, the number of supplies on the design should be minimized if possible. When multiple voltages are required, the die area over which the voltage must be supplied should be minimized. This may require moving I/O buses or IP blocks to different locations on the die.

Once the voltage partitioning is complete, the interaction among the various physical blocks must be considered. The hard IP blocks on the design must be organized in order to minimize top-level interconnect wiring while at the same time respecting I/O placement requirements and the timing budget of the design. Sufficient die area must be reserved for the top-level interconnect, taking into consideration the number of levels of metal that exist on the die, the number of levels of metal utilized in each IP block, free wiring tracks available through each block, and the area of the die that is consumed by power and clock distribution.

Voltage islands

The IBM Blue Logic ASIC Cu-11 product supports voltage islands in the front-end and back-end design methodologies. Previous ASIC products offered voltage islands in the form of either digital or analog hard cores with isolated power supplies. The capability for an SoC designer to freely create voltage islands with customer logic and isolate the supply voltages is a very powerful design capability when automated in an ASIC design system. SoC designers can leverage voltage islands in the ways listed below:

- Power performance management.
- Mixed-voltage IP integration and reuse.
- Standby power management.
- Active power management.
- Voltage isolation.

Voltage islands are handled as RLMs in both front-end and back-end design. All of the design tools, including simulation, synthesis, static timing, power calculation, test insertion, noise analysis, PD optimization, placement, power routing, power bus analysis, and electrical rules checking, must have knowledge of the voltage levels within each island or knowledge that the voltage supply could be disconnected at each level of the voltage island design hierarchy.

The initial offering of voltage island support in Cu-11 includes voltage isolation with off-chip control. **Figure 4** is an example of how power would be brought into a voltage island on a wire bond die. The voltage isolation support includes inter-island communication within the full voltage range of Cu-11. Inter-island communication circuits were specifically designed in the ASIC library to enable this level shifting across voltage islands at different potentials. In addition to the inter-island communication when the voltage islands are powered up, special fencing circuits are provided in the ASIC library to handle voltage island communications when any voltage island is powered down. These force a known state on nets that may connect to other powered islands. All of the voltage island constraints are handled by the design system checking, simulation, synthesis, test insertion, physical design (PD) optimization, placement, and power-routing tools for an efficient ASIC implementation with processing time very close to that for standard hierarchical design.

Electrical analysis

During the SoC design, several electrical aspects receive ongoing attention as the floorplan evolves. The main aspects of interest are the integrity of the power supply and signals. These two concerns can be controlled with correct placement and simultaneous switching analysis.

Power-supply integrity

During the definition of the default power distribution for the die, the power busing *RLC* properties are defined and a model is built. A maximum current density is established, accounting first for uniform current density, then assuming hot spots of various sizes and currents. The dc analysis is performed first, taking into account IR drops and electromigration. Testing of the ac properties includes inductive drops assuming no decoupling and a simple triangular current waveform, consistent with the dc current-density assumptions.

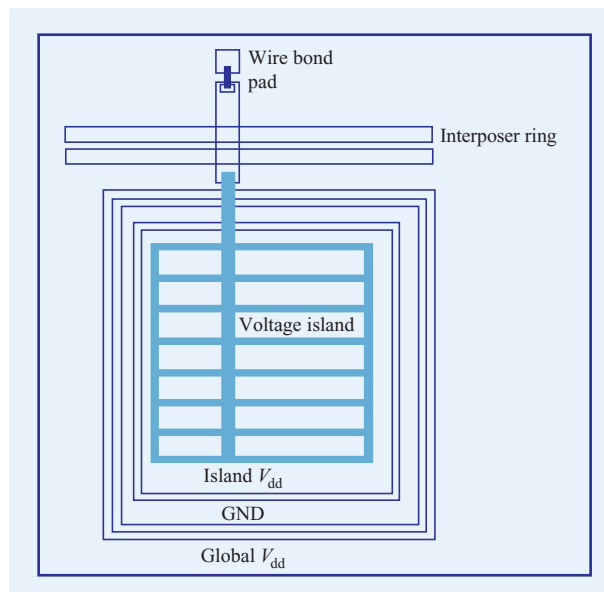


Figure 4

Example of power connection to voltage island on a wire bond image. Reprinted with permission from [6].

When required for custom images, the power busing is incrementally modified by making local changes followed by specific analysis. Special cases such as the clock distribution might exceed the power grid current rating. These are designed and analyzed separately, including their own timing and decoupling analysis. Large predesigned elements such as memories connect to the power grid in a specified manner. They perform their own power-supply analysis on the basis of what they can expect from the die power grid. Other predesigned elements which connect directly to the power bumps that they “own” can perform an independent power analysis.

Analysis of signal integrity is performed using a simplified model of the signal lines. According to the speed required, the package signal wires may be reduced to lumped self and mutual inductances. In more demanding cases, the electrical elements represent only a few millimeters of signal wire, no more than 1/20 the wavelength. In addition to inductance, skin-effect resistance, capacitance, and conductance may be included. The signal return path presents challenges at higher frequencies, as described in [7].

I/O placement is critical to good electrical behavior. Simple guidelines are given, such as creating small I/O placement areas bounded by solder bumps, or a “window,” as shown in **Figure 5**. Though the rules may be simple, they prevent catastrophic placement situations. For instance, a hot spot can be prevented by restricting the

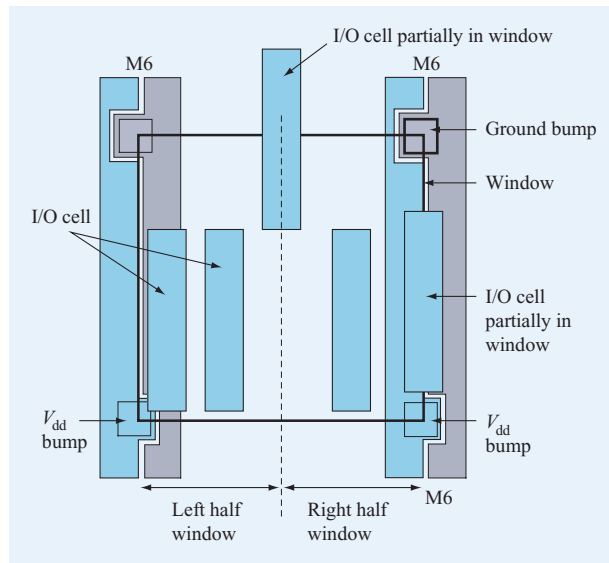


Figure 5

Window used for rules-based placement and verification. Reprinted with permission from [6].

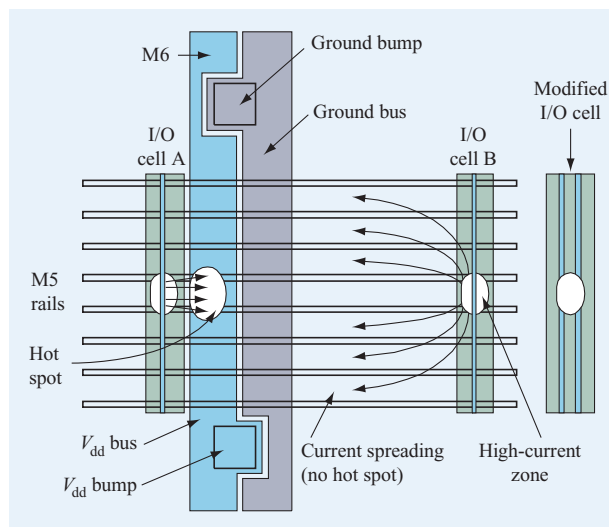


Figure 6

Left, original hot-spot current formation; right, modified cell with resulting current spreading. Reprinted with permission from [6].

I/O placement near the vertical power bus, where the high-current zone of an I/O will be supplied through only a few horizontal power rails, creating potential electromigration concerns (**Figure 6**). Modifications to the I/O cell to distribute the current vertically and placement farther from the vertical power trunks allow current

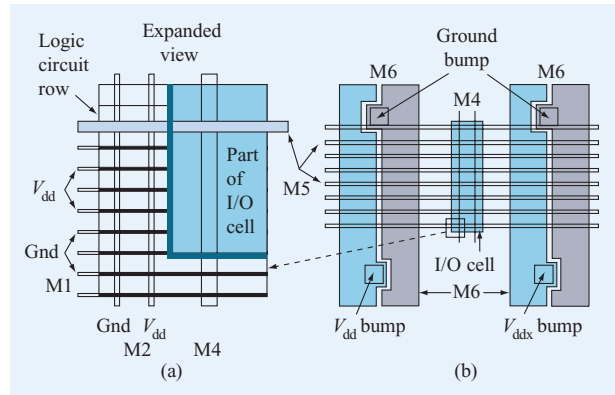


Figure 7

Bus structure and expanded I/O view: (a) Simplification of the power busing, used for placement. (b) Additional simplification, used in verification. Reprinted with permission from [6].

spreading through multiple horizontal rails. The number of I/Os placeable in a window is determined by referring to a table summarizing the I/O properties, including maximum charging currents, average currents, and slew rates.

Reference [11] offers some details of low-level rules for local I/O placement verification. The verification is performed one window at a time. The tool has a window description, as shown in **Figure 7**. The tool extracts the placement of all of the I/Os in a window, and the rules are verified. Failures are flagged with comments. The method is very fast, and it is incorporated in interactive placement tools.

The simplest technology rules are used for global I/O placement legalization. The verification is performed on a complete die. The tool extracts the actual die power grid and builds an *RLC* electrical model with several million elements. The tool extracts the placement of all of the I/Os on the die and inserts equivalent current sources. A fast linear solver computes the IR drops and the grid current densities. The computed values are compared against the basic technology rules, and temperature maps illustrate the results of the analysis. This approach can be generalized to perform ac analysis, using equivalent circuits to represent the package across the solder bumps [12, 13].

SoC routing

Each new deep-submicron ASIC technology reduces the metal-line-spacing design rules, creating more opportunity for line-to-line coupling capacitance. SoC designers are being challenged to continue aggressively designing for performance while trying to avoid line-to-line noise coupling issues as technologies are being scaled. Xrouter,

the IBM signal routing tool, has a wire-spreading capability that has had a significant impact on reducing the line-to-line coupling capacitance-induced noise as well as improved manufacturing yield by reducing line-to-line critical areas. Mandatory design signoff requirements for cores and die require wire spreading to be completed on all ASIC designs.

In addition to wire spreading, the ASIC tools Xrouter and Scorpion (balanced clock- and signal-routing tools) have the capability to use shielded wire types in Cu-11 product. The shielded wire types are an explicit way to isolate or shield a wire, in contrast to spreading, which is a constraint on the router to create the isolation where possible. **Figure 8** shows the shielding scenarios available to SoC designers:

- Standard wire type/no shielding for comparison (signal wire in orange).
- Shield-and-isolate wire type/shield (two shield wires in purple, two isolation wires in green, and the signal wire in orange).
- Shield-only wire type/shield (two shield wires in purple and the signal wire in orange).
- Isolate-only wire type/shield (two isolation wires in green and the signal wire in orange).

Each shielded wire type has a different characteristic amount of blockage, 3D extraction predictability, and total wire capacitance. **Table 3** shows how each of these characteristics affects SoC designs in determining how often and where within the design to use each wire type.

SoC buffer and wire optimization

During floorplanning of SoC designs, the number of connections across large blocks can be minimized but not completely eliminated. The IBM repeater optimization tool, BuffOpt, can automatically select wire sizing, plane selection, or insertion of a repeater for the best solution to address large SoC blockages issues. When BuffOpt identifies a net which can benefit from the thick copper

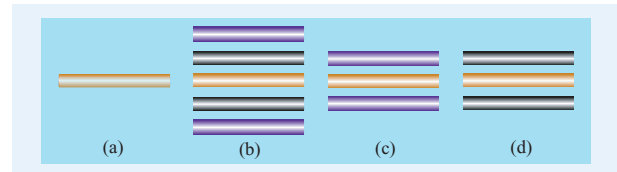


Figure 8

Shielding scenarios (see text).

level, it automatically passes this constraint to the router. Wire sizing and plane selection both have the effect of reducing the resistance and increasing the capacitance of a wire. BuffOpt considers net topology, driver strength, timing constraints, and wire-level parasitics when optimizing nets.

Another approach to solving the large-block SoC timing closure issue is to design into the IP repeater cell holes that would be available to the optimization tools as valid placement locations. Such holes may constitute more optimal locations for repeaters than having to place repeaters completely outside the blockage of large cores in SoC designs. Repeater holes are floorplanned in large hard cores along functional boundaries to provide space for repeaters to enable more flexibility in chip-level timing closure. The methods used to insert repeaters in chip-level repeater bays, as described in [14], are the same as would be used in the hard-core repeater holes. In the IBM Blue Logic SA-27E ASIC technology, the PPC440A4V1 core was designed with repeater holes and used on the SoC chip design shown in **Figure 9**.

Figure 9 illustrates a good example of the physical layout of an SoC ASIC designed within this methodology. The sub-blocks labeled in the figure represent a variety of hard cores, soft cores, and application-specific hierarchical blocks. Many of the sub-blocks are defined with move bounds to allow unconstrained logic to flow within them. Fixed macro blocks, such as SRAMs, are shown as black

Table 3 Comparison of shielded-wire-type characteristics.

Wire type	Total wire capacitance	3D extraction predictability	Blockage	Best application
Isolate only	Least	Least	Shield-only ↕	General noise coupling correction, clock driver to splitter
Shield only	Most	Most	Isolate-only	Critical timing accuracy, best noise coupling reduction
Shield-and-isolate	Intermediate	Intermediate	Most	Performance timing accuracy, clock driver to driver

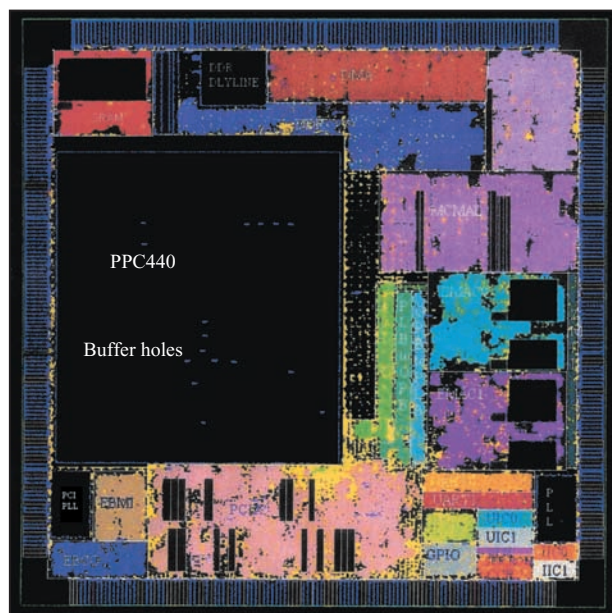


Figure 9

SA-27E SoC chip design (buffer holes enlarged for better visibility).

rectangles within the functional units. I/O circuits and pads form the outer ring at the periphery of the die.

SoC clocking

IBM Blue Logic ASIC clock methodology offers two options to SoC designers. The first option is the structured clock buffer (SCB) approach described in [15]. This approach is targeted for SoC designs that require high-performance clock trees. The key elements of the SCB clock methodology include a clock-planning tool (ClockPro), a clock-optimization tool (ClockDesigner), and a balanced clock-routing tool (Scorpion). The SCB circuit is a four-stage buffer circuit and is physically designed to spread across the chip power grid and mitigate electromigration issues due to the high switching factor and frequency characteristics of high-performance clock trees. Two types of SCB circuits are available to SoC designers for easier floorplanning: horizontal and vertical. The horizontal SCB is short and very wide, while the vertical SCB is very tall and narrow.

The second clock methodology option is the standard distributed clock tree, which uses regular clock buffers and typically more stages to implement a clock tree than does the SCB clock methodology. The standard clock tree methodology has much more flexibility during floorplanning because of the smaller size of the clock buffer circuits, but at the cost of much higher clock skew;

for this reason it is targeted for lower-performance SoC designs.

Placement-driven synthesis

A very important design capability included in the IBM Blue Logic ASIC design system is the placement-driven synthesis (PDS) tool, which significantly reduces the amount of time needed to achieve timing closure on SoC designs by reducing the number of iterations involving floorplanning, synthesis, and layout, as described in [16]. The higher accuracy of placement-based wire estimation clearly has had a major impact on the time it takes to close timing on large SoC designs compared to previous methods involving wire-load-based statistical models. PDS is the integration of three IBM tools: BooleDozer*, a logic synthesis tool; EinsTimer*, a static timing tool; and ChipBench* placement algorithms using a transformal approach, as described in [17]. SoC designers in IBM technologies have the advantage of two modes of using PDS. PDS can be used in refine mode, in which the input is a synthesized and placed netlist, or full placement mode, in which the input is a synthesized netlist.

Summary

The design size and complexity of an SoC design, and the variety of silicon technology features required, have forced the development of an efficient design system architecture to provide the productivity necessary to ensure first-pass success using a reasonably sized design team. These designs introduce at the chip level functional, physical, and electrical challenges previously met only at the board level. Concepts such as multiple voltage supplies, hierarchy, I/O planning, and electrical analysis must be optimized for this environment. These considerations affect the architecture of the physical design system itself, as well as the process flows for the realization of each design.

The construction and execution of the physical design concepts described above resulted in multiple SoC ASIC designs with first-time success. These concepts facilitate the reuse of IP and provide advantages in overall design turnaround time, time-to-market, and resource management.

Acknowledgments

The authors wish to acknowledge the continued contributions of the Technology Development team, the ASIC Product Development team, the ASIC Methodology team, and the EDA Development team to the SoC design system architecture defined in this paper. The authors would like to specifically acknowledge Paul Dunn and Pat Ryan for data volume test-case results.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. T. R. Bednar, R. A. Piro, D. W. Stout, L. Wissel, and P. S. Zuchowski, "Technology-Migratable ASIC Library Design," *IBM J. Res. & Dev.* **40**, No. 4, 377–386 (1996).
2. J. Dreibelbis, J. Barth, Jr., R. Kho, and H. Kalter, "An ASIC Library Granular DRAM Macro with Built-In Self Test," *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1998, pp. 74–75.
3. J. Dreibelbis, J. Barth, Jr., H. Kalter, and R. Kho, "Processor Based Built In Self Test for Embedded DRAM," *IEEE J. Solid State Circuits* **33**, No. 11, 1731–1740 (1998).
4. J. E. Barth, Jr., J. H. Dreibelbis, E. A. Nelson, D. L. Anand, G. Pomichter, P. Jakobsen, M. R. Nelms, J. Leach, and G. M. Belansek, "Embedded DRAM Design and Architecture for the IBM 0.11- μ m ASIC Offering," *IBM J. Res. & Dev.* **46**, No. 6, 675–689 (2002, this issue).
5. M. R. Ouellette, D. L. Anand, and P. Jakobsen, "Shared Fuse Macro for Multiple Embedded Memory Devices with Redundancy," *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2001, pp. 191–194.
6. Patrick H. Buffet, Joseph Natonio, Robert A. Proctor, Yu H. Sun, and Gulsun Yasar, "Methodology for I/O Cell Placement and Checking in ASIC Designs Using Area-Array Power Grid," *IBM MicroNews* **6**, No. 3, 7–9 (2000).
7. A. E. Ruehli and A. C. Cangellaris, "Progress in the Methodologies for the Electrical Modeling of Interconnects and Electronic Packages," *Proc. IEEE* **89**, No. 5, 740–771 (2001).
8. W. D. Becker, J. Eckhardt, R. W. Frech, G. A. Katopis, E. Klink, M. F. McAllister, T. G. McNamara, P. Muench, S. R. Richter, and H. H. Smith, "Modeling, Simulation, and Measurement of Mid-Frequency Simultaneous Switching Noise in Computer Systems," *IEEE Trans. Components, Packaging, Manuf. Technol. Part B* **21**, No. 2, 157–163 (1998).
9. D. J. Herrell and B. Beker, "Modeling of Power Distribution Systems for High-Performance Microprocessors," *IEEE Trans. Adv. Packaging* **22**, No. 3, 240–248 (1999).
10. Alex Waizman and Chee-Yee Chung, "Resonant Free Power Network Design Using Extended Adaptive Voltage Positioning (EAVP) Methodology," *IEEE Trans. Adv. Packaging* **24**, No. 3, 236–244 (2001).
11. P. H. Buffet, J. Natonio, R. A. Proctor, Y. H. Sun, and G. Yasar, "Methodology for I/O Cell Placement and Checking in ASIC Designs," *Proceedings of the IEEE Custom Integrated Circuits Conference*, May 2000, pp. 125–128.
12. S. R. Nassif and J. Kozhaya, "Multi-Grid Methods for Power Grid Simulation," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2000, pp. 457–460.
13. J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "I/O Buffer Placement Methodology for ASICs," *Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems*, 2001, pp. 245–248.
14. C. J. Alpert, G. Gandham, J. Hu, J. L. Neves, S. T. Quay, and S. S. Sapatnekar, "Steiner Tree Optimization for Buffers, Blockages and Bays," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2001, pp. 399–402.
15. K. M. Carrig, N. T. Gargiulo, R. P. Gregor, D. R. Menard, and H. E. Reindel, "A New Direction in ASIC High-Performance Clock Methodology," *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1998, pp. 593–596.
16. D. E. Lackey, "Applying Placement-Based Synthesis for On-Time System-on-a-Chip Design," *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2000, pp. 121–124.
17. W. Donath, P. Kudva, L. Stok, P. Villarrubia, L. Reddy, and A. Sullivan, "Transformational Placement and Synthesis," *Proceedings of the Conference and Exhibition on Design, Automation and Test in Europe*, 2000, pp. 194–201.

Received November 15, 2001; accepted for publication May 7, 2002

Thomas R. Bednar *IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (tbednar@us.ibm.com).* Mr. Bednar received a B.S. degree in electrical engineering from the University of Notre Dame in 1982 and an M.S. degree in electrical engineering from the University of Vermont in 1988. He joined IBM in Essex Junction in 1982 and has worked on the definition, design, and qualification of several generations of ASIC products and function libraries. Mr. Bednar is currently a Senior Technical Staff Member in ASIC Product Development.

Patrick H. Buffet *IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (buffetph@us.ibm.com).* Mr. Buffet received a Diplome d'Ingenieur degree from the Institut National Polytechnique de Grenoble, France. He worked with Texas Instruments and Digital Equipment Corporation before joining IBM in 1992. Mr. Buffet has designed integrated circuits and currently performs electrical analysis of die images and packages. He is a member of the IEEE and has served on technical program and standards committees. He has ten filed patent applications and eight others pending.

Randall J. Darden *IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (rjdarden@us.ibm.com).* Mr. Darden joined IBM in 1983 and has worked in several areas of ASIC design, methodology, and support. He received a B.S. degree in 1982 and an M.S. degree in 1983, both in electrical engineering from Washington State University. He also received the M.S. degree in computer science in 1999 from National Technical University. Mr. Darden is currently a Senior Engineer in the ASIC Layout Services and Software Department; he has interests in ASIC physical design, including hierarchical processing and timing-driven methodologies.

Scott W. Gould *IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (swgould@us.ibm.com).* Mr. Gould received an M.S. degree in electrical engineering from Syracuse University in 1986, and a B.S. degree in electrical engineering from the University of Maine at Orono in 1983. He is an Advisory Engineer working in the ASIC Physical Design Methodology development group, where he is currently the ASIC physical design methodology team leader, with a focus on technology and core development requirements. He has worked in the areas of ASIC library development, physical design methodology, and FPGA design.

Paul S. Zuchowski *IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (paulz@us.ibm.com).* Mr. Zuchowski received a B.S. degree in electrical and computer engineering from Clarkson University in 1989. He then joined IBM in Essex Junction. In 1995, he received an M.S. degree in microelectronic manufacturing from Rensselaer Polytechnic Institute. Mr. Zuchowski has worked on full-chip physical verification, physical model generation methodology, physical design methodology, and ASIC product architecture. He is currently a Senior Engineer in ASIC product development.